Cyclic production in regular robotic cells: A counterexample to the 1-cycle conjecture

Florence Thiard, Nicolas Catusse, Nadia Brauner

Cyclic production in regular robotic cells: A counterexample to the 1-cycle conjecture

Florence Thiard, Nicolas Catusse, Nadia Brauner

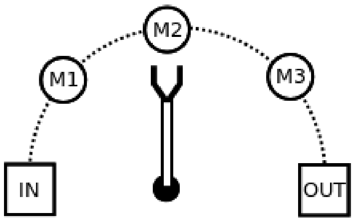
March 26, 2015

1 Linear and circular robotic cells

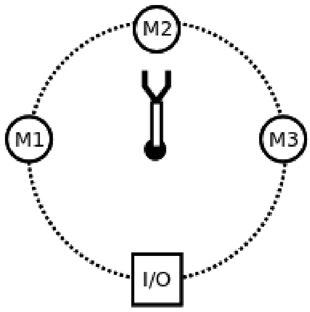
Robotic flow-shops consists in m machines disposed in linear, semi-circular or ciruclar layout, served by a robotic arm. We focus on the case where all the parts to produce are identical. The objective is to maximize the throughput of the cell. A survey on robotic cells can be found in [2].

The robotic cells we consider are composed of m bufferless machines, denoted by $M_1, M_2 \dots M_m$, an entry buffer IN and an exit buffer OUT. The parts must be processed on each of the machines $M_1, M_2 \dots M_m$ in that order. The parts are available in infinite quantity at buffer IN ; likewise, the exit buffer OUT has infinite capacity.

We represent the entry buffer IN and the exit buffer OUT by two auxiliary machines, respectively M_0 and M_{m+1}. In a cell with circular layout, IN and OUT are in the same place: $M_0 = M_{m+1}$. Figure (1a) shows an example of a 3-machine semi-circular cell, while figure (1b) shows a 3-machine circular cell.



(a) semi-circular layout



(b) circular layout

Figure 1: 3-machine robotic cells

We focus on the single-gripper robot, bufferless case, which means the robot and the machines $M_1 \dots M_m$ can only handle one part at a time. Thus, the robot has to be empty to pick up a part from a machine.

The cell operates with unbounded waiting policy (once processed, a part can remain on a machine as long as necessary) and additive travel times.

In this work, we make two further assumptions: we assume that the cell is balanced, meaning that the processing times on machines $M_1 \dots M_m$ are the same, and regular, which means that the distance between two consecutive machines is always the same. In this case, the problem's input consists in four numbers: m the number of machine, p the processing time, δ the travel time between two consecutive machines, and ϵ the loading/unloading time.

In a cell with linear or semi-circular layout verifying these assumptions, the travel time between 2 machines M_i and M_j is $\delta_{i,j} = |i-j|\delta$. In a cell with circular layout, the travel time between 2 machines is defined by the shortest path along the circle, so $\delta_{i,j} = \min(|i-j|, m+1-|i-j|)\delta$. So far, circular layouts have been less studied than linear (or semi-circular) layouts, and some important results on the latter are not applicable to the former.

We consider cyclic robot moves. A k-cycle is a production cycle of exactly k parts: during one iteration of the cycle, k parts enter the cell through M_0, k parts exit the cell through M_{m+1}, and the cells returns to the same state (same machines loaded, same machines empty, and same position of the robot). In particular, 1-cycles are production cycles of 1 part.

To describe the cycles we will use the concept of activities, introduced in [8]. For $i \in \{0 \dots m\}$, activity A_i refers to the following sequence of events:

- The robot unloads a part from M_i;
- The robot travels to M_{i+1};
- The robot loads the part onto M_{i+1}.

A k-cycle can be described as a sequence of activities. The authors in [8] showed that a k-cycle is a sequence in which each activity A_i occurs exactly k times, and there is exactly one occurrence of A_i between A_{i-1} and A_{i+1} (in a cyclic sense), for $i \in \{1 \dots m-1\}$. More specifically a 1-cycle can be described as a permutation of activities $A_0, \dots, A_m$ [11].

A cycle is optimal if it maximizes the throughput rate or equivalently minimizes the cycle time over the number of parts produced in one iteration $\frac{T(C_k)}{k}$. A set S of cycles is dominant if, for any instance, there exists a cycle of S that is optimal.

1-cycles are of a special interest as in a linear cell, the best 1-cycle can be found in polynomial time [7]. However, this result is not applicable to circular cells. Rajapakshe *et al.* in fact proved in [10] that finding the best 1-cycle is NP-hard, for cells with a similar circular-type layout.

2 Dominant set of cycles : the 1-cycle conjecture

In [11], the authors state the following conjecture, and prove it is valid for 2 machine cells:

Conjecture 2.1 (1-cycle conjecture) *The set of 1-cycle is dominant*

For linear, regular, unbounded cells, the authors in [8] and [4] proved that the conjecture is true for 3 machines, but a counterexample for 4-machines is shown in [5]. In the balanced case, the conjecture has been proven true for up to 6 machines [5, 3]. According to [2], the proof can be extended up to 15 machines.

We provide a counterexample to the 1-cycle conjecture for the circular case.

Theorem 2.1 *In a regular balanced unbounded 6-machine cell with circular layout, the 2-cycle*

$$\hat{C} = (A_0 A_3 A_2 A_5 A_1 A_4 A_3 A_6 A_0 A_2 A_5 A_4 A_1 A_6)$$

dominates all 1-cycles for the following instance:

$$\delta = 1 \quad \epsilon = 0 \quad p = 11$$

.

In order to prove Theorem 2.1, we study some classical cycles and establish some necessary properties of dominant cycles. In the following, we always assume that $\epsilon = 0$ and neglect this parameter. As the instance in Theorem 2.1 has $\epsilon = 0$, this assumption is sufficient for the proof.

Since the 2-machine case has been completely solved for both linear and circular layout, we will also always assume that $m \geq 3$.

2.1 Lower bounds on the cycle times

First, we present two classical lower bounds, valid both for linear and circular layout. The formulation is adapted to the regular balanced case.

Property 2.1 *[6] Any k-cycle c verifies*

$$T(c) \geq k(p + 4\delta) \tag{1}$$

This bound is the minimum time between two loadings of the same machine.

Property 2.2 *[9] Any k-cycle c verifies*

$$T(c) \geq k((m+1)\delta + m \min(p, \delta)) \tag{2}$$

Intuitively, if an activity A_i is immediately followed by the subsequent activity A_{i+1}, then the robot waits p time units ; if not, it adds at least δ to its minimum travel time.

Now, we introduce the following notations, for every cycle c:

- $\Delta(c)$ is the total travel time of the robot,
- $d_{i,k}(c)$ is the travel time of the robot between the k-th loading of machine M_i and its subsequent unloading (in a cyclic sense).
- $d_{min}(c) = \min_{i,k}(d_{i,k}(c))$

The following lower bound is actually a lower bound on the cycle-time of a given cycle, depending on this parameter.

Property 2.3 *For any cycle c, the cycle time verifies*

$$T(c) \geq \Delta(c) + \max(0, p - d_{min}(c)) \tag{3}$$

Proof

It is easy to see that $T(c) \geq \Delta(c)$.

Let $(i_0, k_0) = \arg\min_{i,k}(d_{i,k}(c))$. Between the k_0-th loading of M_{i_0} and its subsequent unloading, the robot travels d_{min}, but there must be at least p units of time for the part to be processed. So, if $p \geq d_{min}$, somewhere between the loading and the unloading, the robot must wait (additionally to its travel time) at least $p - d_{min}$ units of time. Hence $T(c) \geq \Delta(c) + \max(0, p - d_{min}(c))$ $\square$

2.2 Some classical cycles

Identity cycle

We call identity cycle (also named uphill permutation or forward cycle in the literature) the cycle $\pi_{id} = (A_0 A_1 ... A_m)$. The cycle time of this cycle is

$$T(\pi_{id}) = (m+1)\delta + mp \tag{4}$$

In this cycle, the robot circles the cell once. Intuitively, this cycle is interesting for instances for which p is much smaller than δ. From the lower bound in Property 2.2, one can immediately derive the following:

Property 2.4 *If $p \leq \delta$, then the identity cycle π_{id} is optimal.*

Downhill cycle

We call downhill cycle (also named reverse cycle) the cycle $\pi_d = (A_0 A_m A_{m-1} ... A_1)$. The cycle time of this cycle is

$$T(\pi_d) = 3(m+1)\delta + \max(0, p - (3m-1)\delta) \tag{5}$$

In this cycle, each spot is visited by the robot three times. Intuitively, this cycle is interesting for instances for which p is much greater than δ. From the lower bound in Property 2.1, one can easily derive the following:

Property 2.5 *If $p \geq (3m-1)\delta$, then the downhill cycle π_d is optimal.*

Odd-Even cycle

In circular cells, a third 1-cycle of particular interest is the odd-even cycle, defined as such:

For m even,

$$\pi_{oe} = (A_0 A_2 A_4 \dots A_m A_1 A_3 A_5 \dots A_{m-1})$$

And for m odd,

$$\pi_{oe} = (A_0 A_2 A_4 \dots A_{m-1} A_1 A_3 A_5 \dots A_m).$$

In this cycle, the robot circles around the cell twice: the first time performing even activities (thus loading odd machines), the second time performing odd activities (thus loading even machines). The cycle time for the odd-even cycle is

$$T(\pi_{oe}) = 2(m+1)\delta + \frac{2\alpha - 1}{\alpha} \max(0, p - (m+1)\delta) \qquad \text{with } m = \begin{cases} 2\alpha & \text{if } m \text{ even} \\ 2\alpha - 1 & \text{if } m \text{ odd} \end{cases}$$

Proof

Consider α consecutive iterations of π_{oe}. The robot travels $2(m+1)\alpha\delta$, circling the cell 2α times. Let us follow the path of one same part in the cell.

During the first loop, the part M_1 is loaded with A_0. It is then unloaded during the second loop with A_1. Between the loading and the unloading, the robot travels $(m+1)\delta$ and the piece must stay on M_1 at least p, so the robot must wait $\max(0, p-(m+1)\delta)$. Similarly, during any loop, the part is loaded on a machine, then unloaded and taken to the next machine during the next loop, and the robot must wait $\max(0, p-(m+1)\delta)$. If m is odd, the part exits on the last loop. If m is even, it is loaded on M_m during the last loop.

Eventually, for this one part, the robot must wait at least $(2\alpha-1)\max(0, p-(m+1)\delta)$ over α iterations of the cycle. Therefore,

$$T(\pi_{oe}) \geq 2(m+1)\delta + \frac{2\alpha-1}{\alpha}\max(0, p-(m+1)\delta)$$

Now, let us call w_j^i the waiting time of the robot at machine M_j during the i-th iteration of cycle π_{oe}, and $W^i = (w_1^i, \ldots, w_m^i)$. Let $a = p-(m+1)\delta$.

Case 1: m odd $\pi_{oe} = (A_0A_2A_4\ldots A_{m-1}A_1A_3A_5\ldots A_m)$ The waiting times at iteration i are :

$$\left\{\begin{array}{rcl} w_{2j}^i & = & \max(0, a - \sum\limits_{k=1+j}^{\alpha} w_{2k-1}^{i-1} - \sum\limits_{k=1}^{j-1} w_{2k}^i) \\ w_{2j-1}^i & = & \max(0, a - \sum\limits_{k=j}^{\alpha-1} w_{2k}^i - \sum\limits_{k=1}^{j-1} w_{2k-1}^i) \end{array}\right.$$

If $a \leq 0$ then all waiting times w are zero, and the cycle time is $2(m+1)\delta$.

If not, we can easily check that the vector $W_0 = (\frac{a}{\alpha}, \ldots \frac{a}{\alpha})$ is a fixed point of W. The corresponding cycle time is $2(m+1)\delta + \frac{2\alpha-1}{\alpha}a$

Case 2: m even $\pi_{oe} = (A_0A_2A_4\ldots A_mA_1A_3A_5\ldots A_{m-1})$

Similarly, the waiting times at iteration i are

$$\left\{\begin{array}{rcl} w_{2j}^i & = & \max(0, a - \sum\limits_{k=1+j}^{\alpha} w_{2k-1}^{i-1} - \sum\limits_{k=1}^{j-1} w_{2k}^i) \\ w_{2j-1}^i & = & \max(0, a - \sum\limits_{k=j}^{\alpha} w_{2k}^i - \sum\limits_{k=1}^{j-1} w_{2k-1}^i) \end{array}\right.$$

If $a \leq 0$ then $W = 0$, and the cycle time is $2(m+1)\delta$.

If not, we can check that the vector $W_0 = (0, \frac{a}{\alpha}, \ldots \frac{a}{\alpha})$ is a fixed point of W. The corresponding cycle time is $2(m+1)\delta + \frac{2\alpha-1}{\alpha}a$

Finally,

$$T(\pi_{oe}) = 2(m+1)\delta + \frac{2\alpha-1}{\alpha}\max(0, p-(m+1)\delta)$$

$\square$

2.3 Best 1-cycle

In this section, we focus on the problem of the best 1-cycle. We seek the region of optimality (in the sense of optimality over all 1-cycles) of the three cycles presented in the previous section, and show some necessary properties of optimal 1-cycles in the remaining region. As the best 1-cycles are known for $m = 3$ and $m = 4$ (see [1]), in the following we assume $m \geq 5$.

In particular, for 6-machine cells satisfying our initial assumptions, we show that $\{\pi_{id}, \pi_{oe}, \pi_d\}$ form a set a dominant 1-cycles.

To make the following proofs more readable, we will use this notation for activities:

$$A_i = A_{i \bmod (m+1)}$$

2.3.1 Another lower bound...

Additionally to the lower bounds presented in Section 2.1 , we establish the following lower bound on 1-cycles:

Property 2.6 *If $p \geq \delta$, and π is a 1-cycle with $\pi \neq \pi_{id}$, then*

$$T(\pi) \geq 2(m+1)\delta \tag{6}$$

Proof Let $\pi \neq \pi_{id}$ be a 1-cycle different from the identity cycle.

Case 1: If π contains a sequence A_iA_j with $j \neq i+2$ and $j \neq i+1$, then we have

$$T(\pi) \geq (m+1)\delta + (m-1)\min(p, \delta) + 2\delta \geq 2(m+1)\delta$$

as it takes the robot at least 2δ to travel from M_{i+1} to M_j while performing A_iA_j.

Case 2: If not, then π contains a sequence A_iA_{i+2}, and every other 2-element sub-sequence in π can be written A_jA_{j+1} or A_jA_{j+2}. Then, we know that:

- The robot always travels in the same direction, forward. With $m \geq 3$, The shortest circular path from a machine M_j to M_{j+1} never requires to go backward.
- The robot travels at least 2 times between M_{i+1} and M_{i+2}. Once loaded, while performing activity A_{i+1} and once empty, while performing the sequence A_iA_{i+2}.

Hence the robot circles the cell at least twice:

$$T(\pi) \geq 2(m+1)\delta$$

$\square$

2.3.2 Regions of optimality for classical cycles

In this section, we determine some values of parameter p, depending on δ, for which one of the 3 cycles π_{id}, π_{oe} and π_d dominates 1-cycles.

Property 2.7

(i) if $p \leq \frac{m+1}{m}\delta$, then the identity permutation π_{id} is dominates 1-cycles.

(ii) *if* $\frac{(m+1)}{m}\delta \leq p \leq (m+1)\delta$*, then the odd-even cycle* π_{oe} *dominates 1-cycles.*

(iii) *if* $p \geq 3(m-1)\delta$*, the downhill permutation* π_d *is dominates1-cycles.*

Proof
Property (iii) follows directly from bound 1.
For $p \leq \delta$, then from bound 2 π_{id} is optimal.
For $\delta \leq p \leq \frac{m+1}{m}\delta$, we have

$$T(\pi_{id}) = (m+1)\delta + mp \leq 2(m+1)\delta$$

and, from Property 2.6, for any 1-cycle π:

$$T(\pi) \geq 2(m+1)\delta \geq T_{\pi_{id}}$$

which implies (i).
Finally, for $\frac{(m+1)}{m}\delta \leq p \leq (m+1)\delta$, we have

$$T(\pi_{oe}) = 2(m+1)\delta$$

$$T(\pi_{id}) = (m+1)\delta + mp \geq 2(m+1)\delta$$

and for any 1-cycle $\pi \neq \pi_{id}$, from property 2.6:

$$T(\pi) \geq 2(m+1)\delta$$

which implies (ii). □

Now remains the case of instances where $(m+1)\delta \leq p \leq (3m-1)\delta$. This is the issue we address in the next section.

2.3.3 Necessary properties of optimal 1-cycles

We now assume that $(m+1)\delta \leq p \leq (3m-1)\delta$. We establish some properties a 1-cycle must satisfy in order to do better than both the odd-even cycle and the downhill permutation in this region.
Let π^* be a 1-cycle verifying, for $(m+1)\delta \leq p \leq (3m-1)\delta$,

$$\begin{cases} T(\pi^*) & < & T(\pi_{oe}) \\ T(\pi^*) & < & T(\pi_d) = 3(m+1)\delta \end{cases}$$

Property 2.8 π^* *satisfies the following property:*

(i) $2(m+1)\delta < \Delta(\pi^*) < 3(m+1)\delta$

(ii) π^* *doesn't contain any sequence* A_iA_{i+1} *with* $i \neq m$

(iii) $d_{min}(\pi^*) > \Delta(\pi^*) - \frac{3\alpha-2}{2\alpha-1}(m+1)\delta$

We will prove this property through a series of claims.

Claim 2.1 *π^* contains no sequence A_iA_{i+1} with $i \neq m$.*

Proof First, π^* contains at most one sub-sequence of two consecutive activities (except from the sequence A_mA_0), otherwise $T(\pi^*) \geq (m+1)\delta + 2p \geq 3(m+1)\delta$.

Let us assume now that it exists one and only one $i \neq m$ so that A_iA_{i+1} is a sub-sequence of π^*. We can already deduce that

$$T(\pi^*) \geq (m+1)\delta + (m+1)\delta + (m-1)\delta \tag{7}$$

Case 1: $\pi^* = A_0A_1 \dots A_m$

In this case, either π^* contains the subsequence $A_{m-1}A_2$, or there exist indices $j \notin \{m, 0\}$ and $k \notin \{0, 1\}$ so that $A_{m-1}A_j$ and A_kA_2 are two distinct sub-sequences of π^*. In both cases, the robot travels at least an additional 2δ compared to (7).

Case 2: $\pi^* = A_0 \dots A_{m-1}A_m$

In this case, either π^* contains the subsequence $A_{m-2}A_1$, or there exist indices $j \notin \{m-1, m\}$ and $k \notin m, 0$ so that $A_{m-2}A_j$ and A_kA_1 are two distinct sub-sequences of π^*. In both cases, the robot travels at least an additional 2δ.

Case 3: $\pi^* = A_0 \dots A_iA_{i+1} \dots A_m$, with $1 \leq i \leq m-2$

In this case, either π^* contains the subsequence $A_{i-1}A_{i+2}$, or there exist indices $j, k \notin \{i, i+1\}$ so that $A_{i-1}A_j$ and A_kA_{i+2} are sub-sequences of π^*. In the second case, the robot travels an additional 2δ. In the first case, π^* can be written in one of the following ways :

$$\pi^* = A_0 \dots A_iA_{i+1}SA_{i-1}A_{i+2} \dots A_m \tag{8}$$

$$\pi^* = A_0 \dots A_{i-1}A_{i+2}SA_iA_{i+1} \dots A_m \tag{9}$$

Where S is an activity sequence, including the empty sequence.

The robot already travels an additional δ while performing $A_{i-1}A_{i+2}$. In order to have no additional travel time, the sequence $A_{i+1}SA_{i-1}$ in (8) or $A_{i+2}SA_i$ in (9) must be a subsequence(in the cyclic sense) of the following sequences :

$$\begin{cases} A_0A_2A_4 \dots A_{m-1} \text{ or } A_1A_3A_5 \dots A_m & \text{if } m \text{ is odd} \\ A_0A_2A_4 \dots A_mA_1A_3A_5 \dots A_{m-1} & \text{if } m \text{ is even} \end{cases}$$

which is impossible as S cannot contain A_0 or A_m. So, the robot travels at least an additionnal 2δ compared to (7)

Case 4: $\pi^* = \dots A_iA_{i+1} \dots$ and π^* does not contain A_mA_0

In this case, either π^* contains the sub-sequence A_mA_2, either there exist indices $j, k \notin \{i, i+1\}$ and $l \neq 0$ so that $A_{i-1}A_j$, A_kA_{i+2} and A_mA_l are subsequences of π^*, with at least 2 of these subsequences distinct. In both cases, the robot travels at least an additional 2δ.

In all cases, we have $T(\pi^*) \geq (m+1)\delta + (m+1)\delta + (m-1)\delta + 2\delta = 3(m+1)\delta$ □

Claim 2.2 *The travel time $\Delta(\pi^*)$ associated with π^* verifies $\Delta(\pi^*) \geq 2(m+1)\delta$.*

Proof This is a consequence of 2.1. As π^* contains no sequence of the form A_iA_{i+1} with $i \neq m$, there is two possibilities:

Case 1: There exists a sub-sequence A_iA_j with $j \neq i+2$ in π^*. Then the travel time of the robot is at least $(m+1)\delta + m\delta + \delta = 2(m+1)\delta$

Case 2: π^* contains only subsequences of the form A_iA_{i+} and maybe A_mA_0. Similarly to the proof of 2.6, we can say that the robot always travels forward, and circles the cell twice: $\Delta(\pi^*) \geq 2(m+1)\delta$. □

Claim 2.3 *The travel time $\Delta(\pi^*)$ associated with π^* verifies $\Delta(\pi^*) > 2(m+1)\delta$.*

Proof Let us assume that $\Delta(\pi^*) = 2(m+1)\delta$. We already know that π^* can have no subsequences A_iA_{i+1} except for A_mA_0.

Case 1: π^* doesn't contain the sequence A_mA_0. Then all sub-sequences have to be of the form A_iA_{i+2} :the only possible cycle is of the form $A_0A_2A_4 \dots A_mA_1A_3 \dots A_{m-1}$, which is π_{oe} if m is even, and not possible if m is odd.

Case 2: π^* contains the sequence A_mA_0. π^* contains only one sequence of the form A_iA_{i+3} or A_iA_{i-1}. The others are of the form A_iA_{i+2}. π^* contains a sequence $A_{m-1}A_j$ with $j \neq 0$, and a sequence A_lA_1 with $l \neq m$. Thus, π^* must contain the sequence $A_{m-1}A_1$. The only possible cycle is of the form $A_0A_2A_4 \dots A_{m-1}A_1A_3 \dots A_m$, which is π_{oe} if m is odd, and not possible if m is even. □

Claim 2.4 *The travel time $\Delta(\pi^*)$ associated with π^* verifies $\Delta(\pi^*) < 3(m+1)\delta$*

Proof Otherwise, we would have $T(\pi^*) \geq 3(m+1)\delta = T(\pi_d)$ □

Claim 2.5 *$d_{min}(\pi^*)$ verifies $d_{min}(\pi^*) > \Delta\pi - \frac{3\alpha-2}{2\alpha-1}(m+1)\delta$*

Proof Assume δ is fixed, and call $T_c(p)$ the cycle time of any cycle c depending on p. We assume $p \leq (3m-1)\delta$.
Let's call $p_{int} = \frac{3\alpha-1}{2\alpha-1}(m+1)\delta$ the value of parameter p for which π_d and π_{oe} have the same cycle time.

$$T_{\pi_{oe}}(p_{int}) = T_{\pi_d}(p_{int}) = 3(m+1)\delta$$

Let f the affine function with coefficient 1 verifying

$$f(p_{int}) = T_{\pi_{oe}}(p_{int}) = T\pi_d(p_{int}) = 3(m+1)\delta$$

.

The expression of f is

$$f(p) = p + \frac{3\alpha-2}{2\alpha-1}(m+1)\delta$$

For $(m+1)\delta \leq p \leq p_{int}$,

$$f(p) \geq T_{\pi_{oe}}(p) \geq T_{pi_d}(p)$$

For $p_{int} \leq p \leq (3m-1)\delta$,

$$f(p) \geq T_{\pi_d}(p) \geq T_{pi_{oe}}(p)$$

Let us introduce

$$g(p) = \Delta(\pi^*) + \max\left[0, p - (\Delta(\pi^*) - \frac{3\alpha-2}{2\alpha-1}(m+1)\delta)\right]$$

.

If $d_{min}(\pi^*) \leq \Delta(\pi^*) - \frac{3\alpha-2}{2\alpha-1}(m+1)\delta$, then, from 2.3 we can deduce:

$$T_{\pi^*}(p) \geq \Delta(\pi^*) + \max(0, p - d_{min}(\pi^*)) \geq g(p) \geq \min(T_{\pi_{oe}}(p), T_{\pi_d}(p))$$

.

So, $d_{min}(\pi^*) > \Delta(\pi^*) - \frac{3\alpha-2}{2\alpha-1}(m+1)\delta$. □

2.4 Dominant set of 1-cycles

We know that 1-cycles are permutations of activities. Given a 1-cycle, it is easy to check if it complies with the conditions exposed in Property 2.8. Of course this can only be done for small numbers of machines as the number of 1-cycles is exponential.

For $6 \leq m \leq 8$, we computed all 1-cycles and found that no cycle matches the conditions in Property 2.8.

Proposition 2.1 *For a regular balanced cell, with $6 \leq m \leq 8$ and $\epsilon = 0$, the set $\{\pi_0, \pi_{oe}, \pi_d\}$ dominates all 1-cycles.*

2.4.1 Conclusion

Finally, we can prove Theorem 2.1.

Let us consider a 6-machine circular cell, with $p = 11$, $\delta = 1$, $\epsilon = 0$. From Proposition 2.1, we know that $\{\pi_0, \pi_{oe}, \pi_d\}$ dominates all 1-cycles.

For these parameters, one has

$$T(\pi_{id}) = 7 + 66 = 73 \tag{10}$$

$$T(\pi_{oe}) = 14 + \frac{5}{3}4 = 20 + \frac{2}{3} \tag{11}$$

$$T(\pi_{dh}) = 21 \tag{12}$$

We just need to show that $\frac{T(\hat{C})}{k} \leq T(\pi_{oe}) = 20 + \frac{2}{3}$.

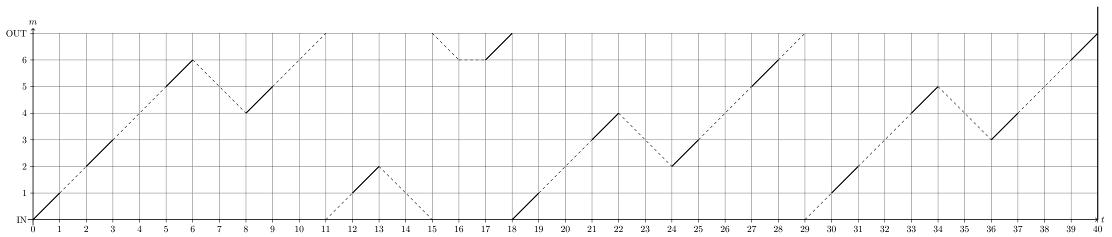


Figure 2: An iteration of C, for $p = 11$ and $\delta = 1$

During one iteration of $\hat{C}$, the robot travels $\Delta(C) = 39$.

For more commodity, let us rewrite C by simple rotation:

$$(A_0A_2A_5A_4A_1A_6A_0A_3A_2A_5A_1A_4A_3A_6)$$

We have $d_{6,1} = d_{2,2} = 10$, and for all other (i,k), $d_{i,k} \geq 11$. As the first unloading of M_6 takes place between the first loading of M_2 and its subsequent unloading, the robot only needs to wait one unit of time, before unloading M_6 (see Figure 2).

Thus, we have $T(\hat{C}) = 39 + 1 = 40$

$$\frac{T(\hat{C})}{2} = 20 < T(\pi_{oe})$$

In conclusion, $\hat{C}$ is strictly better than all 1-cycles on this instance. We proved Theorem 2.1, thus showing that the 1-cycle conjectures is false for 6 machine.

References

[1] N. Brauner. *Ordonnancement dans des cellules robotisées.* Thèse de doctorat, Université Joseph Fourier, Grenoble, France, 1999.

[2] N. Brauner. Identical part production in cyclic robotic cells: Concepts, overview and open questions. *Discrete Applied Mathematics*, 156(13):2480–2492, 2008.

[3] N. Brauner, Y. Crama, and G. Finke. One-unit production cycles in balanced robotic cells. In *Proceedings IEPM'01, International Conference on Industrial Engineering and Production Management*, pages 508–518, Quebec City, Canada, 2001.

[4] N. Brauner and G. Finke. On a conjecture about robotic cells: new simplified proof for the three-machine case. *Journal of Information Systems and Operational Research - INFOR: Scheduling in Computer and Manufacturing Systems*, 37(1):20–36, 1999.

[5] N. Brauner and G. Finke. Cycles and permutations in robotic cells. *Mathematical and Computer Modelling*, 34(5-6):565–591, 2001.

[6] Y. Crama and J. van de Klundert. Cyclic scheduling of identical parts in a robotic cell. *Operations Research*, 45(6):952–965, 1997.

[7] Y. Crama and J. van de Klundert. Robotic flowshop scheduling is strongly NP-complete. In O.J. Vrieze W.K. Klein Haneveld and L.C.M. Kallenberg, editors, *Ten Years LNMB*, pages 277–286, CWI Tract 122, Amsterdam, The Netherlands, 1997.

[8] Y. Crama and J. van de Klundert. Cyclic scheduling in 3-machine robotic flow shops. *Journal of Scheduling*, 2:35–54, 1999.

[9] M. Dawande, C. Sriskandarajah, and S. Sethi. On throughput maximization in constant travel-time robotic cells. *Manufacturing and Service Operations Management*, 4(4):296–312, 2002.

[10] T. Rajapakshe, M. Dawande, and C. Sriskandarajah. Quantifying the impact of layout on productivity: An analysis from robotic-cell manufacturing. *Operations Research*, 59(2):440–454, 2011.

[11] S. P. Sethi, C. Sriskandarajah, G. Sorger, J. Blazewicz, and W. Kubiak. Sequencing of parts and robot moves in a robotic cell. *International Journal of Flexible Manufacturing Systems*, 4:331–358, 1992.